CHAPTER 1

Pattern and Anomaly Discovery in Data

Bart Baesens, Wouter Verbeke, Johannes De Smedt, Jochen De Weerdt, Hans Weytjens

Corresponding author: Bart.Baesens@kuleuven.be

1.1 CHAPTER OBJECTIVES

In this chapter, you learn to

- understand the workings of association and sequence rules;
- evaluate association and sequence rules in terms of support, confidence, lift and conviction;
- extend and post-process association rules;
- apply association and sequence rules;
- discover anomalies using break point analysis, peer group analysis, isolation forest, Local Outlier Factor, and one-class SVMs;

1.2 INTRODUCTION

In this chapter, we zoom in on pattern and anomaly discovery in data. First, this chapter aims to equip you with the understanding and skills to explore the underlying relationships in datasets through association and sequence rules. We discuss metrics such as support, confidence, lift, and conviction which quantify rule strength from various perspectives. Post processing and applications of both association and sequence rules are also extensively covered.

In a second part, we elaborate on anomaly detection. Techniques like break point analysis, peer group analysis, isolation forests, Local Outlier Factor, and one-class SVMs will be discussed to identify deviations that might indicate critical insights or anomalies in data. These methods open new avenues for understanding data behaviors that deviate from the norm, offering valuable opportunities in, e.g., fraud detection or rare event modeling.

This chapter serves as a comprehensive guide to both foundational and advanced techniques in unsupervised learning for pattern discovery. By integrating theoretical knowledge with practical applications, you learn to harness the power of data analytics to uncover meaningful patterns and identify significant anomalies that could otherwise go unnoticed.

1.3 ASSOCIATION RULES

1.3.1 Problem Statement

Association rules aim at detecting frequently occurring patterns or relationships between items. Remember that this is an example of descriptive analytics or unsupervised learning, where there is no real target or dependent variable for which to optimize. A first example is market-basket analysis, whereby the goal is to analyze which products or services are frequently bought together. This information can then be used for product bundling or shelf organization. Another example is analyzing which web pages are frequently visited together. This can be useful for optimizing web site design. It can also be used for text analytics to study which terms often co-occur in a text document. This can then be used by text summarization tools. Finally, associations between course electives can be used to optimize university time tabling.

Association rules typically start from a database D of transactions t_p. Every transaction has a transaction identifier and a set of items that are selected from all possible items. In a market-basket analysis context, the transaction ID is the purchase ID and the items are the products that were purchased. In a web analytics context, the transaction ID is the visit ID and the items are the web s that were visited. In a text analytics context, the transaction ID is the document ID and the items are the words in the document. In a university time tabling context, the transaction ID is the student ID and the items are the electives chosen by the student. An association rule is then an implication of the form

$$X \rightarrow Y \tag{1.1}$$

where both X and Y are a subset of all possible items I. The intersection between X and Y is empty ($x \cap Y = \emptyset$). X is often referred to as the rule antecedent and Y the rule consequent. Some examples are as follows:

- If a customer has a car loan and car insurance, then the customer has a checking account in 80% of the cases
- If a customer buys spaghetti, then a customer buys red wine in 70% of the cases

Beer and diapers.
The story goes that in the 1990s, Walmart discovered through data mining (as it was then called) that young fathers often bought beer when shopping for diapers in the evening. Though it has never been truly verified, it is commonly used an example in textbooks covering association rule mining.

In Table 1.1 you can see an example of a transactions database in a market basket analysis setting. Obviously, not every transaction should have the same amount of items.

1.3.2 Support and Confidence

It is very important to be aware of the fact that association rules are statistical rules, in the sense that they are typically only true in a majority of the cases. Hence, measures are needed to quantify the strength of the association. Two key measures are the support and the confidence. The support of an itemset is the percentage of total transactions in the database that contains the itemset, or the rule $X \rightarrow Y$ has support s if $100s\%$ of the transactions in D contain $X \cup Y$. It can be calculated by dividing the number of transactions that support X and Y by the total number of transactions.

$$support(X \cup Y) = \frac{\text{number of transactions supporting } X \cup Y}{\text{total number of transactions}} \tag{1.2}$$

Transaction	Items
1	beer, milk, diapers, baby food
2	coke, beer, diapers
3	cigarettes, diapers, baby food
4	chocolates, diapers, milk, apples
5	tomatoes, water, apples, beer
6	spaghetti, diapers, baby food, beer
7	water, beer, baby food
8	diapers, baby food, spaghetti
9	baby food, beer, diapers, milk
10	apples, wine, baby food

Table 1.1: Example Transaction Database.

A frequent itemset is then an itemset for which the support is higher than a prespecified threshold, which is referred to as minsup.

The association rule $X \to Y$ has confidence c if $100c\%$ of the transactions in D that contain X also contain Y. In other words, the confidence is the conditional probability of the rule consequent Y, given the rule antecedent X. It can be calculated as the support of X and Y, divided by the support of X

$$confidence(X \to Y) = P(Y|X) = \frac{support(X \cup Y)}{support(X)} \tag{1.3}$$

Let's illustrate the calculation of the support and confidence with an example. Let's reconsider the transaction database depicted in Table 1.1. The itemset baby food, diapers, and beer occurs in transactions 1, 6, and 9, giving a support of 3 out of 10 or 30%. Based on this itemset, various association rules can now be derived. Let's pick one as an example: Baby food and diapers → beer. To calculate the confidence of this association rule, we first need to calculate the support of baby food and diapers. This itemset occurs in transactions 1, 3, 6, 8 and 9. Out of these five transactions, three also include beer, giving a confidence of 3 out of 5 or 60%.

1.3.3 Association Rule Mining

Mining association rules from a transactions database is essentially a two-step process as follows:

- **Step 1**: Identification of all item sets having support above minsup, i.e., 'frequent' item sets;
- **Step 2**: Discovery of all derived association rules having confidence above minconf

As said before, both minsup and minconf need to be specified beforehand by the data analyst. The first step is typically performed using the Apriori algorithm [Agrawal and Ramakrishnan, 1994]. The Apriori property states that every subset of a frequent item set is frequent as well, or conversely, every superset of an infrequent item set is infrequent. This implies that candidate item sets with k items can be found by pairwise joining frequent item sets with $k - 1$ items and deleting those sets that have infrequent subsets. Thanks to this property the number of candidate subsets to be evaluated can be decreased which will substantially improve the performance of the algorithm since less databases passes will be required.

Let's illustrate this with an example. Consider the transactions database depicted in Table 1.2 containing only four transactions. Suppose we set the minimum support to 50%. We then find 4 frequent item sets with 1 item as listed in Table 1.3. We then consider all itemsets with 2 items by pairwise merging the frequent itemsets of size 1. This gives us the itemsets listed in Table 1.4. For each of them, we count the support and contrast this against the minimum support of 50%. This gives us 4 frequent itemsets of size 2 as shown in Table 1.5. We continue this reasoning. However, we make sure to apply the Apriori property. For example,

TID	Items
100	apples, diapers, rolex
200	beer, diapers, baby food
300	apples, beer, diapers, baby food
400	beer, baby food

Table 1.2: Transaction database for association rule mining.

Item Sets	Support
{apples}	50%
{beer}	75%
{diapers}	75%
{baby food}	75%

Table 1.3: Association rule mining: step 1.

Item Sets	Support
{apples,beer}	25%
{apples,diapers}	50%
{apples,baby food}	25%
{beer,diapers}	50%
{beer,baby food}	75%
{diapers,baby food}	50%

Table 1.4: Association rule mining: step 2.

Item Sets	Support
{apples,diapers}	50%
{beer,diapers}	50%
{beer,baby food}	75%
{diapers,baby food}	50%

Table 1.5: Association rule mining: step 3.

Item Sets	**Support**
{beer,diapers,baby food}	50%

Table 1.6: Association rule mining: step 4.

	Tea	**Not tea**	**Total**
Coffee	150	750	900
Not coffee	50	50	100
Total	200	800	1000

Table 1.7: Lift of an association rule.

let's say we merge itemsets apples, diapers and beer, diapers to apples, beer, diapers. We already know that this can never give us a frequent itemset since the itemset apples, beer is not frequent, hence, we don't have to consider it. By applying the Apriori property, there is only 1 candidate itemset of size 3 that remains to be considered and that is the one with the items beer,diapers,baby food as depicted in Table 1.6. We can then count its support and indeed find that it is a frequent itemset.

Once the frequent item sets have been found, the association rules can be generated in a straightforward way by considering all possible itemset subsets as rule antecedents and their complement as rule consequents. Consider again the frequent itemset baby food, diapers, beer as found in the transaction database of Table 1.1. The following association rules can be derived with corresponding confidence:

- diapers, beer → baby food [confidence= 75%]
- baby food, beer → diapers [confidence= 75%]
- baby food, diapers → beer [confidence= 60%]
- beer → baby food and diapers [confidence= 50%]
- baby food → diapers and beer [confidence= 43%]
- diapers → baby food and beer [confidence= 43%]

Note that since the confidence is the ratio of two support calculations (see Equation 1.3), it can be efficiently calculated by re-using the support numbers obtained during the Apriori step. If the minconf is set to 70%, only the first two association rules will be kept for further analysis.

As clear by now, when mining association rules, both the minimum support, minsup, and the minimum confidence, minconf, need to be specified in advance by the user. Determining the optimal levels of both thresholds is a difficult task. Setting minsup too low will lead to a combinatorial explosion of the number of candidate-itemsets. Otherwise, setting it too high will miss some important association rules of rare but interesting items. Hence, choosing optimal levels for both parameters should be done carefully and preferably in co-operation with a business expert typically after having tried various values and inspecting the impact thereof on the rule generated.

1.3.4 Lift and Conviction

Consider the example from a supermarket transactions database depicted in Table 1.7. Let's now consider the association rule Tea → Coffee. The support of this rule is 100/1000 or 10%. The confidence of the rule is: 150/200 or 75%. At first sight, this association rule seems very appealing given its high confidence. However, closer inspection reveals that the prior probability of buying coffee equals 900/1000 or 90%. Hence, a customer who buys tea is less likely to buy coffee than a customer about whom we have no information. The lift, also referred to as the interestingness measure, takes this into account by incorporating the prior probability of the rule consequent as follows:

$$Lift(X \rightarrow Y) = \frac{support(X \cup Y)}{support(X)support(Y)} \tag{1.4}$$

A lift value less (larger) than 1 indicates a negative (positive) dependence or substitution (complementary) effect. In our example, the lift value equals 0.89, which clearly indicates the expected substitution effect between coffee and tea.

The conviction of an association rule is calculated as:

$$Conviction(X \rightarrow Y) = \frac{1 - support(Y)}{1 - confidence(X \rightarrow Y)} \tag{1.5}$$

It basically compares the frequency of the consequent being incorrect if the rule was independent on its antecedent to the frequency of the rule being incorrect. It ranges between 0 and infinity. A higher conviction implies a higher dependence of Y on X and hence that the rule is more reliable. A conviction value of 1 occurs if $support(Y)$ equals $confidence(X \rightarrow Y)$ so the occurrence of X basically has no impact on the occurrence of Y. In case $confidence(X \rightarrow Y)$ equals 1, the conviction becomes infinity. For our example in Table 1.7, the conviction of the association rule Tea → Coffee equals: $\frac{1-90\%}{1-0{,}75} = 0, 4$. confirming the counter intuitive relationship since typical conviction values range from 1 to infinity, with 1 indicating independence and values greater than 1 indicating increasing degrees of dependency.

Both lift and conviction are considered complementary evaluation criteria when assessing association rules.

1.3.5 Association rule extensions

The standard support-confidence framework only looks for association rules containing a specified set of items. However, the absence of certain items may also yield important information with respect to the consequent. A straightforward way to accommodate this is to include additional columns indicating the absence of all items in a transaction. Running the Apriori-algorithm on this transformed data set would yield the desired association rules. Obviously, this will undoubtedly further explode the number of association rules further amplifying the need for post processing. Alternatives using contingency tables and Chi-squared analysis have also been suggested in the literature [Silverstein et al., 1998].

So far, we have discussed boolean association rules which only pay attention to whether an item is present in a transaction or not. Quantitative association rules also take into account the quantities of the items in a transaction. One way to mine these is by partitioning each quantitative attribute into a set of intervals, which may overlap, and map the problem to a boolean association rules problem [Srikant and Agrawal, 1996].

A taxonomy specifies a hierarchical, usually tree based, organisation between the items in a transactional database. Think about a product level taxonomy in a retail setting such as milk, diary products, drinks, food. Association rules can then be mined either across or within hierarchical levels of the taxonomy. Obviously, for higher levels, higher minimum support tresholds can be set [Ramakrishnan and Agrawal, 1995].

1.3.6 Post Processing Association Rules

Bart Baesens's first paper.
As a funny anecdote, happy to share that my (= Bart Baesens) first paper ever written was on post processing association rules [Baesens et al., 2000b]. I proud myself of the fact that it keeps on getting cited, even up to this very day. ;-)

The support-confidence based association rule mining process often yields a large number of rules, making it hard for the business user to select the interesting ones. This is particularly the case for data sets whose attributes are highly correlated. The sheer multitude of generated rules often clouds the perception of the users. Hence, post processing of association rules is a key activity [Baesens et al., 2000a]. Rightful assessment of the usefulness of the generated output introduces the need to effectively deal with different forms of rule redundancy and rules being plainly uninteresting.

First, one could filter out the trivial rules that contain already known and obvious patterns such as buying spaghetti and spaghetti sauce, mortgage and home insurance, viewing Rocky III and Rocky IV. This is re-assuring since if they would not occur amongst the results it would be highly suspicious. Hence, finding these obvious patterns serves as nice validation of the association rule mining exercise.

Next, the rules can be ranked in terms of any of the measures discussed above (support, confidence, lift, conviction). The inspection of the rankings can be facilitated by means of two- or three-dimensional visualisations. On-Line Analytical Processing (OLAP) visualisation facilities can be very handy as this facilitates identifying the most interesting rules in a user-friendly way. This is an exercise which is typically conducted in close collaboration between the data scientist and the business user.

Rule pruning essentially amounts to the elimination of association rules because they are redundant or simply prove to be uninteresting. A first example concerns rule subsumption where two rules have the same consequent but one contains additional conditions in its antecedent. Consider the following example:

- **R1**: baby food, diapers → beer [confidence= 72%]
- **R2**: diapers → beer [confidence= 70%]

Clearly, whenever rule **R1** holds rule **R2** will also hold, hence **R1** is subsumed by **R2**. The question is whether **R1** should be kept. This depends upon how much the confidence is increase because of the extra rule condition (diapers in our case). If this is below a certain threshold, say 5%, it can be pruned which is the case in our example.

Rule templates can also be used to prune rule bases. These are general rule skeletons describing structure of the 'interesting' -rules. Each rule template describes what attributes should occur as antecedents or consequents in a rule. An example of a rule template is the following: Any → Beer. This template specifies the desire to look for associations having Beer as their consequent and any other item as their antecedent. Templates can be either inclusive or restrictive. To be interesting, a rule has to match an inclusive template. If it matches a restrictive template it is considered to be uninteresting and pruned away.

Finally, one can also consider the economic impact (e.g., profit, cost, margin) of the association rules. An association rule with some low cost items in its antecedent and a profitable consequent deserve special attention. Moreover, it could e.g. be that some association rules are generated by transactions occurring during a specific time-frame because of a promotional campaign or a specific season. Associations between ski-boots and ski-pants may be more prevalent during the winter than during the summer. By looking at the profit and time distributions of the association rules, one may try to cluster the rules into packets as shown in Figure 1.1.

1.3.7 Association Rule Applications

Association rules can be used in various ways. In a market based analysis context, they can be used to detect which products are frequently bought together. Obviously, this has important implications for targeted marketing such as deciding on the next best offer, product bundling, store layout, shelve organization and catalog design.

Let's assume we have our association rule: diapers and baby food → beer. This pattern can be leveraged in various ways. We can put all products together to make sure they are always purchased jointly. Another option is to put them far apart such that customers may purchase additional items in between. We can package all items and may be add a poorly selling item to it. We can see what happens when we raise the

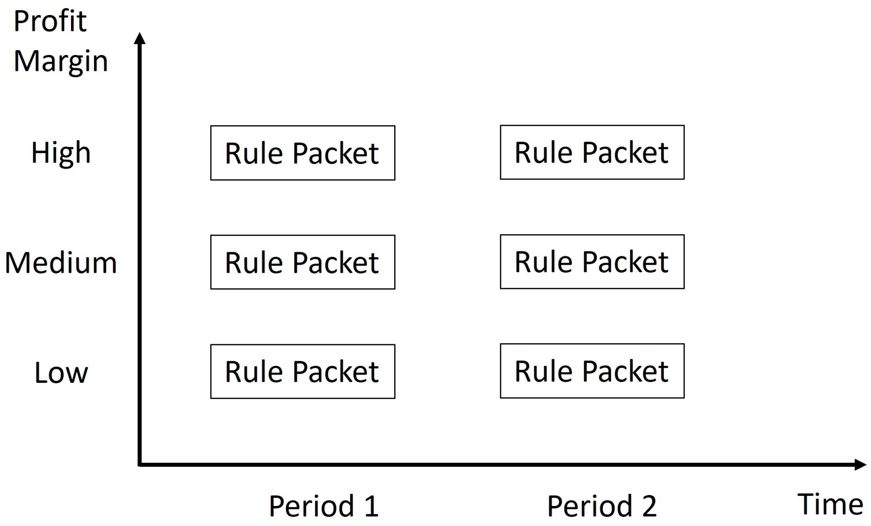


Figure 1.1: Clustering association rules.

price on one and lower it on the other. There is no need to advertise all three together since they are already frequently purchase. In fact, there are plenty of marketing actions we can do based on this information.

Association rules can also be used to build recommender systems that help people make decisions based on preferences. Recommender systems are used in e-business to recommend items, in e-learning to recommend courses and in search and navigation to recommend links. They are commonly used by companies such as Amazon, eBay and Netflix.

Association rules can be used in text analytics to see what terms or concepts frequently co-occur in database of documents. They can also be used in web analytics to find out which web pages are frequently visited together which in turn can help identifying customer journeys. They can also be used in medicine. Association rules can identify sets of genes that frequently co-express under certain conditions. This can help in understanding gene regulatory networks or pathways. In DNA, RNA, or protein sequences, association rule mining can help identify sequence motifs or patterns that frequently occur together, which may be critical for better understanding biological functions.

1.4 SEQUENCE RULES

1.4.1 Problem Statement

Given a database D of customer transactions, the goal of mining sequence rules is to find the maximal sequences among all sequences that have certain user-specified minimum support and confidence. Important to note here is that, contrary to association rules that work with sets, the order of the items in a sequence is important. An example could be a sequence of Web visits in a Web analytics setting: Home → Electronics → Cameras and Camcorders → Digital Cameras → Shopping cart → Order confirmation → Return to shopping. A transaction time or sequence field is now included in the analysis. While association rules are concerned with what items appear together at the same time (intra-transaction patterns), sequence rules are concerned about what items appear at different times (inter-transaction patterns).

Consider the data depicted in Table 1.8 concerning site visits in a Web analytics setting. The letters A, B, C, . . . refer to Web pages and the sessions to web visits. A sequential version can then be obtained as follows:

- Session 1: A, B, C

Session ID	Web page	Sequence
1	A	1
1	B	2
1	C	3
2	B	1
2	C	2
3	A	1
3	C	2
3	D	3
4	A	1
4	B	2
4	D	3
5	D	1
5	C	1
5	A	1

Table 1.8: Example data set for sequence rule mining.

- Session 2: B, C
- Session 3: A, C, D
- Session 4: A, B, D
- Session 5: D, C, A

We can now calculate the support and confidence as with association rules. Consider the sequence rule A →C. We can now calculate the support in two ways. One approach would be to calculate the support whereby the consequent can appear in any subsequent stage of the sequence. Here, the support becomes 2/5 (40%). Another approach would be to consider only sessions where the consequent appears right after the antecedent. Here, the support becomes 1/5 (20%). A similar reasoning can now be followed for the confidence which can then be 2/4 (50%) or 1/4 (25%), respectively.

Sequences can be identified using the Generalized Sequential Pattern (GSP) algorithm.which works very similarly to the Apriori algorithm discussed above. More specifically, it only considers candidate sequences of lenght k if all its subsequences of size $k - 1$ have been identified as frequent in previous iterations.

Sequences can also be evaluated in terms of lift and conviction. The lift measures how much more often a sequence A → B occurs than would be expected if A and B were statistically independent. Similarly, the conviction indicates how often B is dependent upon A.

1.4.2 Sequence Rule Applications

Understanding customer journeys and the resulting customer experience is of key competitive importance to many firms nowadays [Baesens and De Caigny, 2022]. More specifically, a customer journey encompasses all the steps a customer goes through from identifying a need until buying a product or service. Figure 1.2 provides an example of a simplified and somewhat overly stylised customer journey in a mortgage setting. The journey starts from clicking a Facebook Ad and either ends with a stop or accepted offer event.

Here you can see an example of a resulting customer journey. Customer journey analysis serves various business purposes. It can be used to get a clear and comprehensive picture of the overall process and highlight process deficiencies such as excessive processing times, deadlock situations, circular references, and unwanted customer leakage, among others. It can also be used to verify if the process is compliant with both internal and external regulations.

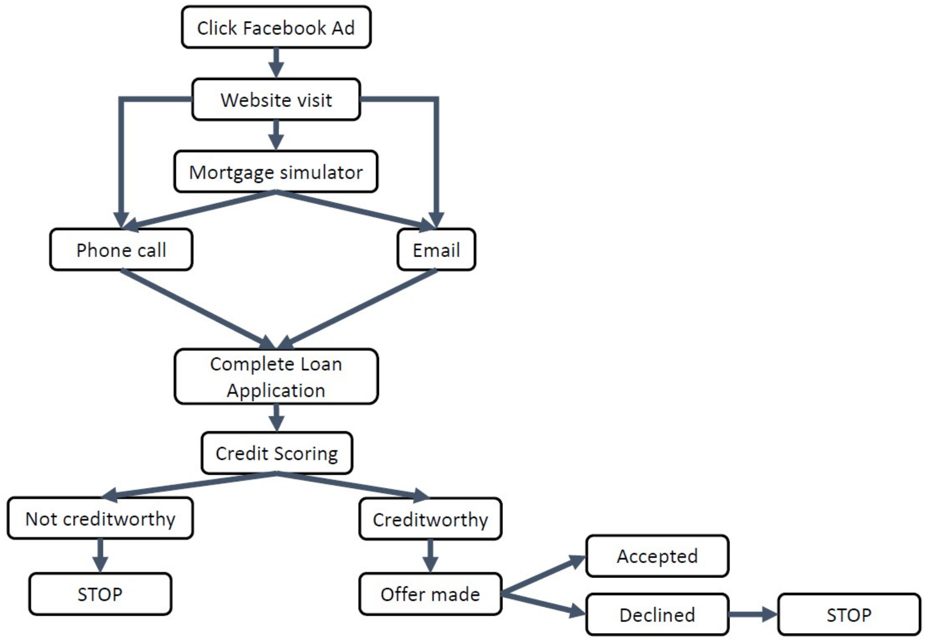
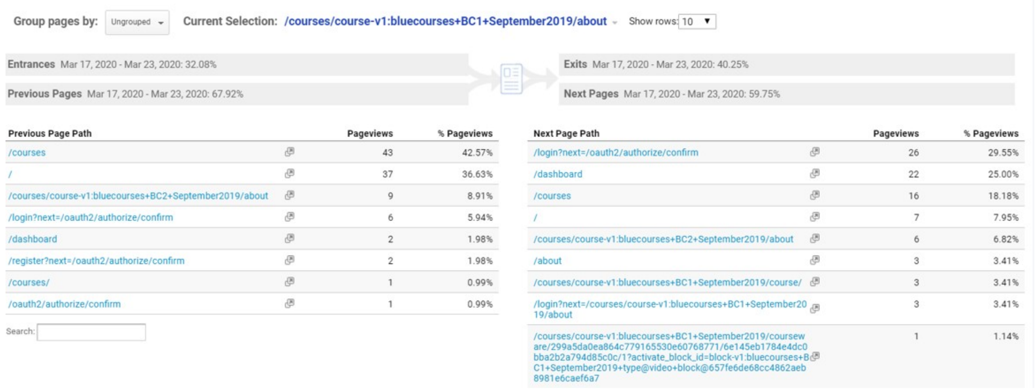


Figure 1.2: Example Customer Journey for mortgages.

Sequence rules have been extensively used for navigation analysis in Web analytics. The idea of navigation analysis is to understand how users navigate through a web site and whether they can easily find what they are looking for. A first way of doing navigation analysis is path analysis, which is based on the analysis of frequent navigation paths. The key question to be answered here is: from a given page, which other pages do users or groups of users visit next in x% of the times, which essentially reflects upon the corresponding sequence rule confidence. We obviously hereby assume that the users follow a linear path which is not always the case since a user may go back or forth, have different tabs open, etc.

In Figure 1.4 you can see path analysis illustrated in Google Analytics. In the red rectangle, you can see the about page of the BlueCourses Basic Credit Risk Modeling course (see `https://www.bluecourses.com/`). Below you can see how many people directly entered through that page, about 32%, and how many were on a previous page before, about 68%. You also see how many exited after having seen that page, around 40%, and how many went on to another page, about 60%. In the table below, to the left, you can see the pages people viewed before the page considered, and to the right, the next page in their path. You can click on any page you see in the report, on the left or right, to then analyze that page – the report will automatically update. You also have the option to click on the name of the page you're currently analyzing in the red rectangle and then search for another page.



Current Selection: /courses/course-v1:bluecourses+BC1+September2019/about

Entrances Mar 17, 2020 - Mar 23, 2020: 32.08%
Previous Pages Mar 17, 2020 - Mar 23, 2020: 67.92%

Exits Mar 17, 2020 - Mar 23, 2020: 40.25%
Next Pages Mar 17, 2020 - Mar 23, 2020: 59.75%

Previous Page Path	Pageviews	% Pageviews
/courses	43	42.57%
/	37	36.63%
/courses/course-v1:bluecourses+BC2+September2019/about	9	8.91%
/login?next=/oauth2/authorize/confirm	6	5.94%
/dashboard	2	1.98%
/register?next=/oauth2/authorize/confirm	2	1.98%
/courses/	1	0.99%
/oauth2/authorize/confirm	1	0.99%

Search:

Next Page Path	Pageviews	% Pageviews
/login?next=/oauth2/authorize/confirm	26	29.55%
/dashboard	22	25.00%
/courses	16	18.18%
/	7	7.95%
/courses/course-v1:bluecourses+BC2+September2019/about	6	6.82%
/about	3	3.41%
/courses/course-v1:bluecourses+BC1+September2019/course/	3	3.41%
/login?next=/courses/course-v1:bluecourses+BC1+September2019/about	3	3.41%
/courses/course-v1:bluecourses+BC1+September2019/courseware/299a5da0ea864c779165530e60768771/6e145eb1784e4dc0bba2b2a794d85c0c/1?activate_block_id=block-v1:bluecourses+BC1+September2019+type@video+block@657fe6de68cc4862aeb8981e6caef6a7	1	1.14%

Figure 1.3: Path Analysis in Google Analytics.

In Figure 1.4 you can see another example of path analysis in Google Analytics. It is called a users flow report and is a graphical representation of the paths users follow through a site, from the source, through the various pages, and where along their paths they exited the site. At the top left, you can see that we are interested in looking at differences between navigation paths of visitors from different countries. You can easily use other criteria such as acquisition channel, new versus return visitor, marketing campaign, type of browser, etc. In this report, you see nodes which represent pages and connections which represent navigating from one page to another. The sizes of the nodes and connections are proportional to the number of visitors. The red colored parts represent leakage or visitors dropping out after having seen that page. The report is interactive; allowing you to highlight different navigation paths to see the flow for those sections without losing sight of the overall navigation picture.

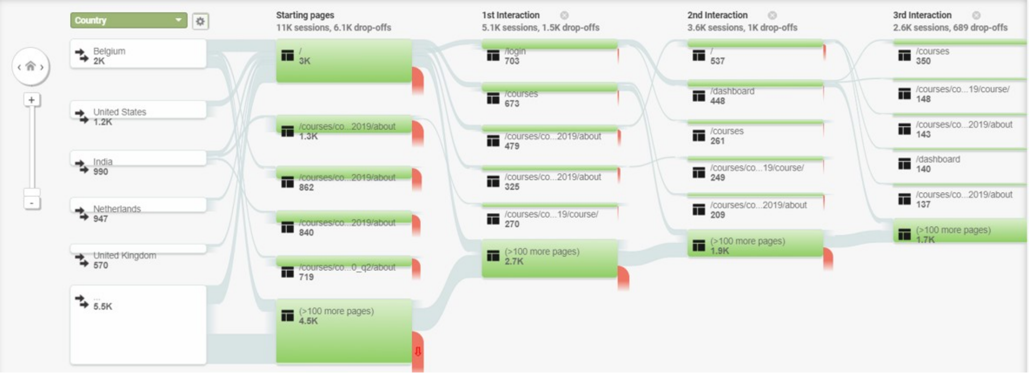
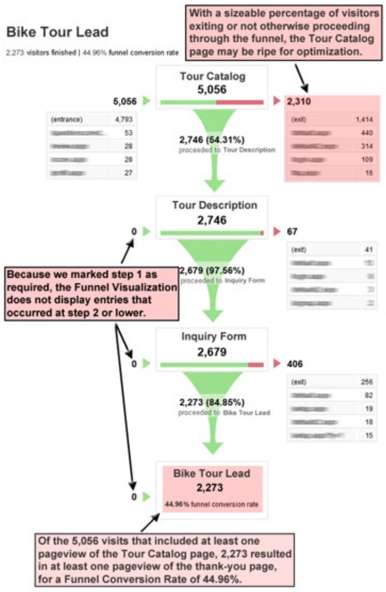


Figure 1.4: Users flow report in Google Analytics.

Finally, in Figure 1.5 you can see an example of a funnel plot (taken from `https://neilpatel.com/blog/conversion-funnel-survival-guide`). It concerns a web site selling personalized bike tours and a conversion is defined as a collected bike tour lead. The first step is selecting a bike tour from the tour catalog. To the left of the first step in this funnel, you can see the entrance or referral pages, to the right you can see the exit pages. You can see that out of the 5056 that visited your tour catalog, 2746 moved on to the next stage in the funnel which is the tour description. There is quite a bit of leakage after this first step, so the Tour Catalog page may be ready for optimization. Do note that since step 1 of the funnel was defined as required, we do not have entries that occur at lower steps. From the Tour Description page, you can see how many move on to the inquiry form page and from there onwards to the Bike Tour lead page which represents a conversion. You can see that 2273 visitors converted which represents a 44.96% funnel conversion rate.

Impact of GDPR on Google Analytics.
The EU General Data Protection Regulation (GDPR) significantly impacted Google Analytics by requiring explicit user consent for data collection and processing. Google had to modify its service to comply with GDPR requirements, including data anonymization, shorter data retention periods, and enhanced data deletion capabilities. The regulation classified IP addresses as personal data, forcing Google Analytics to offer IP anonymization features. Additionally, website owners using Google Analytics became obligated to update their privacy policies, implement cookie consent mechanisms, and establish data processing agreements. This led many European organizations to reconsider their use of Google Analytics, with some data protection authorities ruling that the service's data transfers to US servers violated GDPR principles, ultimately contributing to Google's development of Google Analytics 4 with enhanced privacy features often at the cost of lower quality of the analytical reports and insights.



With a sizeable percentage of visitors exiting or not otherwise proceeding through the funnel, the Tour Catalog page may be ripe for optimization.

Because we marked step 1 as required, the Funnel Visualization does not display entries that occurred at step 2 or lower.

Of the 5,056 visits that included at least one pageview of the Tour Catalog page, 2,273 resulted in at least one pageview of the thank-you page, for a Funnel Conversion Rate of 44.96%.

Figure 1.5: Funnel plot.

1.5 ANOMALY DETECTION

1.5.1 Problem Statement

Anomalies represent observations that deviate from the overall pattern in the data. The deviation could be due to various reasons. One of them is poor data quality with the anomaly being due to a data entry mistake. However, anomalies can also represent valid observations which just happen to be different from the rest because of their exceptional characteristics. Fraud detection is undoubtedly one of the most popular applications of anomaly detection where anomalies can represent unusual credit card transactions, insurance claims, or banking transactions indicating money laundering or other financial crimes. Other examples of anomaly detection applications are in healthcare (e.g., rare diseases), environmental sciences (e.g., climate change) and IoT sensor (e.g., device malfunctioning).

1.5.2 Basic Methods

Some very basic methods can already be useful for detecting anomalies. When only interested in univariate anomalies, a simple minimum/maximum calculation can do the job. Alternatively, a histogram (as discussed in Chapter **??**) can be considered. A visual inspection of unexpected values (e.g., age below 10 or above 90) can already be very revealing. *z*-scores are another option to detect univariate outliers (see also Chapter **??**). The Mahalanobis distance is a straightforward extension for identifying multivariate outliers. It is defined as:

$$d(\mathbf{x}, \mu)_{Mahalanobis} = \sqrt{(\mathbf{x} - \mu)^T \mathbf{\Sigma}^{-1} (\mathbf{x} - \mu)} \quad (1.6)$$

It measures the distance between an observation $\mathbf{x}$ and the population mean μ taking into account the data variability using the inverse of the covariance matrix, ■$^{-1}$. Anomalies can then be identified by sorting all observations in terms of their Mahalanobis distance from the mean and considering the highest values first. You can see this illustrated in Figure 1.6 where the blue observations represent normal data points, the green observation the mean μ, and the red observations the anomalies with their Mahalanobis distance depicted next to them.

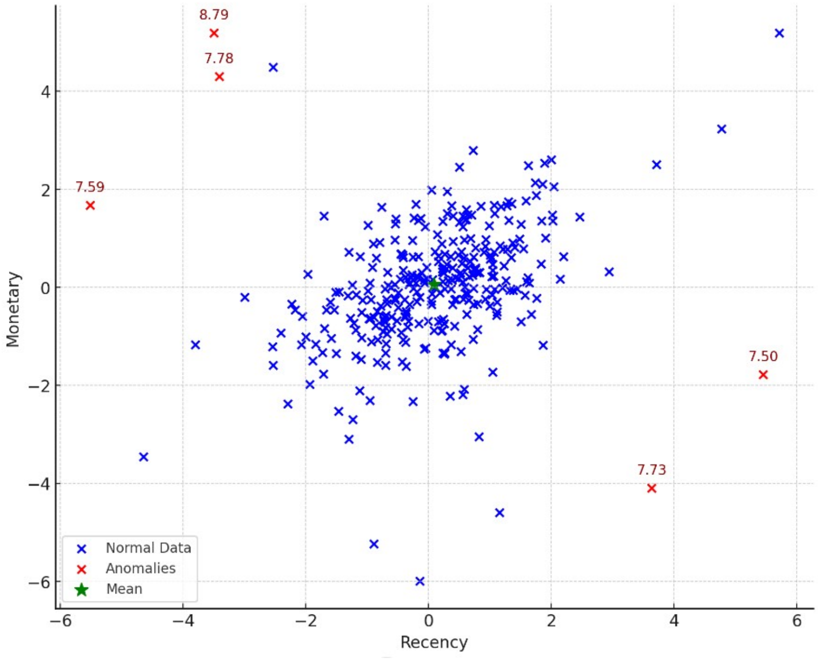


Figure 1.6: Mahalanobis distance for finding anomalies.

1.5.3 Break Point Analysis

Breakpoint analysis is popular fraud detection method for transactional data such as credit card transactions. More specifically, it is an intra-account fraud detection method. You can see it illustrated in Figure 1.7. A breakpoint indicates a sudden change in account behavior that merits further inspection. The method starts from defining a fixed-time window. This time window is then split into old and new parts. The old part represents the local model or profile against which the new observations will be compared. In the example, the time window was set to 24 transactions where 20 transactions were in the local model, and four transactions were used for testing. A Student's t test can be used to compare the averages of the new and old parts. Observations can then be ranked according to their values of the t statistic. Note that break point analysis works at the account-level so does not build profiles by looking at other accounts.

1.5.4 Peer Group Analysis

Peer group analysis is another interesting method for detecting outliers in transactional data. It was introduced by Bolton and Hand in 2001. It starts by defining a peer group. A peer group is a group of accounts that behave similarly in the past to the target account. When the behavior of the latter starts to deviate substantially from its peers, an anomaly can be signaled. Peer group analysis focusses on local instead of global anomalies depending, obviously depending upon the number of peers to consider. The method is especially useful to monitor behavior over time, for example, in time series analysis. Table 1.9 shows a time series of credit card amounts spent by various customers.

Assume that the target account is customer Bart with time series $\mathbf{y_1}, \mathbf{y_2}, \ldots \mathbf{y_{n-1}}$. The aim is now to verify whether the amount spent at time n, $\mathbf{y_n}$, is anomalous. Peer group analysis then starts by identifying the k peers of Bart. Suppose that the k-most similar customers are Wouter, Hans, Johannes and Jochen. To see whether $\mathbf{y_n}$is an outlier, a t score can be calculated as follows:

$$\frac{y_n - \overline{x_{1:k,n}}}{s} \tag{1.7}$$

where $\overline{x_{1:k,n}}$ represents the average of the amounts spent by the k peers at time n and s is the corresponding

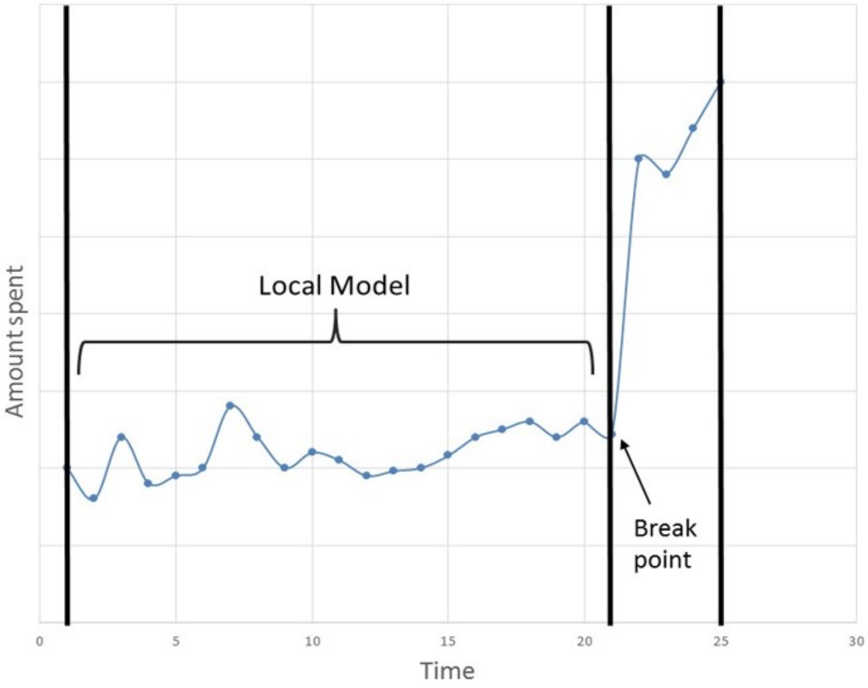


Figure 1.7: Break point group analysis.

	Time				
	1	2		$n-1$	n
	...				
Wouter	$x_{m,1}$	$x_{m,2}$	...	$x_{m,n-1}$	$x_{m,n}$
Hans	$x_{k,1}$	$x_{k,2}$		$x_{k,n-1}$	$x_{k,n}$
	...	...	...	...	...
Johannes	$x_{2,1}$	$x_{2,2}$	...	$x_{2,n-1}$	$x_{2,n}$
Jochen	$x_{1,1}$	$x_{1,2}$	...	$x_{1,n-1}$	$x_{1,n}$
Bart	$\mathbf{y_1}$	$\mathbf{y_2}$	...	$\mathbf{y_{n-1}}$	$\mathbf{y_n}$

Table 1.9: Transactions data for peer group analysis.

standard deviation. This calculation can then be done for each of the customers at time *n*. They can then be sorted in terms of the *t* score and the ones with the highest scores can be further inspected to see whether they are really anomalous.

Figure 1.8 shows a visual representation of peer group analysis. You can see that, at time 25, one account starts to seriously deviate from its peers. A key advantage of peer group analysis, when compared to breakpoint analysis, is that it tracks anomalies by considering inter-account instead of intra-account behavior. For example, if you were to compare transaction amounts for a particular account with previous amounts on that same account (intra-account), then the spending behavior during Christmas will definitely be flagged as anomalous. By considering peers instead (inter-account), this problem is avoided.

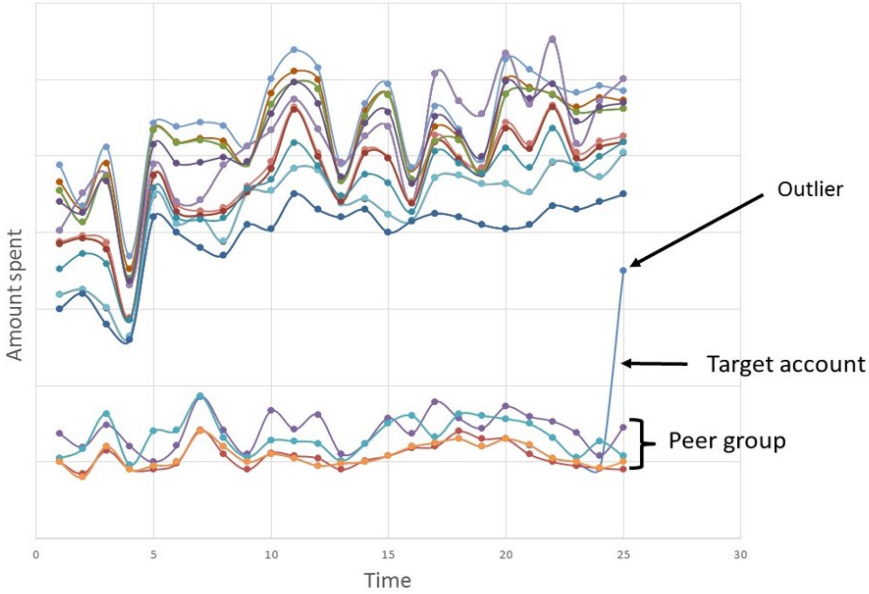


Figure 1.8: Peer group analysis.

David Hand.
Break point and peer group analysis were both introduced by David Hand. David Hand is a prominent British statistician and emeritus professor at Imperial College London. As former president of the Royal Statistical Society (2008-2009), he made significant contributions to classification, data mining, and measurement theory. He developed innovative statistical methodologies including the Hand-Till M measure and wrote influential books like ”Measurement Theory and Practice” (2004). His research spans finance, medicine, and official statistics, earning him an OBE (Officer of the British Empire) in 2013 for his services to research and innovation.

1.5.5 Isolation Forests

The idea of Isolation forest, or iForests, is to train an ensemble or forest of binary trees to detect anomalies [Liu et al., 2008]. The algorithm works as shown in Algorithm 1. An ensemble of T (e.g., 100) trees is created. The first step is to get a sample of the data of size ψ . It then randomly picks a variable and then chooses a random value between its minimum and maximum. Next, a binary split is created using that value. This tree growing process is repeated until the binary tree is complete or when each node has either only one observation or the maximum tree height is reached. The intuition behind the algorithm is that anomalies will be detected using only a few splits whereas normal observations will need a lot of splits to get isolated as they are situated in high density regions. This is illustrated in Figure 1.9. As you can see, the anomaly highlighted in red situated at the right of the graph is already found with the first binary split on

Algorithm 1 Isolation Forest

```
Input: Dataset $D$, Subsample size $\psi$, Number of trees $T$
Output: Isolation Forest $F$
for $i = 1$ to $T$ do
    $D_i \leftarrow$ Randomly sample $\psi$ observations from $D$
    $T_i \leftarrow$ ConstructIsolationTree($D_i, \psi$)
    Add $T_i$ to $F$
end for
procedure ConstructIsolationTree($D, \psi$)
    Randomly select a variable $x$
    Randomly select a split value $v$ between the minimum and maximum values of $x$
    Split the data into two subsets based on $v$
    Recursively apply the process to each subset
    if Each observation is isolated or maximum tree height is reached then
        return Leaf Node
    else
        return Internal Node
    end if
end procedure
```

the $X1$ variable.

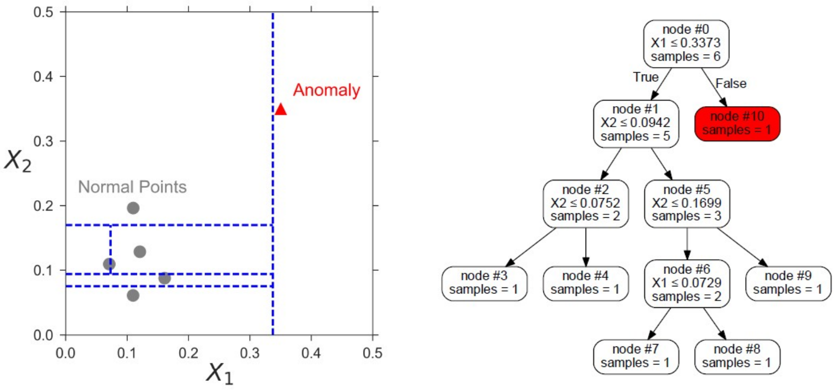


Figure 1.9: Isolation forest [Stripling, 2018].

Since anomalies are more susceptible to isolation, an anomalous observation is expected to have a shorter path length than a normal observation when it traverses a tree from the root to a leaf node. Hence, the isolation score $s(x, n)$ for an observation x from a data set with n samples is calculated as:

$$s(x, n) = 2^{-\frac{E(h(x))}{c(n)}} \tag{1.8}$$

where $E(h(x))$ is the average path length of the observation x, averaged across all trees of the iForest ensemble and $c(n)$ is the average path length of an unsuccessful search (i.e., when a search for a value reaches a leaf node without finding the value) in an iForest tree built using n nodes. As such $c(n)$ quantifies the average depth of an isolation tree and is used to normalize the path lengths in the iForest, allowing the isolation score $s(x, n)$ to be comparable across different datasets and forest sizes. To summarize, if $E(h(x))$ is small compared to $c(n)$, it is more likely that the observation is an anomaly. More specifically, if $E(h(x))$ approximates 0, then $s(x, n)$ will be close to 1 and x can be considered as an anomaly. When the average path length, $E(h(x))$, is very large, $s(x, n)$ will approximate zero which implies that the observation is situated in a high density region of the data and as such is not an anomaly. Hence, higher values of $s(x, n)$ correspond to more likely anomalies. Note that for the exact computation of $E(h(x))$ and $c(n)$, we refer to [Liu et al., 2008].

iForest for workers' compensation claims fraud.
In [Stripling et al., 2018], we propose the iForestCAD approach that computes conditional anomaly scores in fraud detection. iForestCAD performs anomaly detection on well-defined data partitions that are created on the basis of selected numeric attributes and distinct combinations of values of selected nominal attributes. In this way, the resulting anomaly scores are computed with respect to a reference group of interest, thus representing a meaningful score for domain experts. Given that anomaly detection is performed conditionally, this approach allows detecting anomalies that would otherwise remain undiscovered in unconditional anomaly detection. We demonstrate the usefulness of our proposed approach on real-world workers' compensation claims received from a large European insurance organization. The iForestCAD approach was greatly accepted by domain experts for its effective detection of fraudulent claims.

1.5.6 Local Outlier Factor

Local Outlier Factor is an anomaly detection algorithm based on the density of the local neighborhood. Consider the data depicted in Figure 1.10.

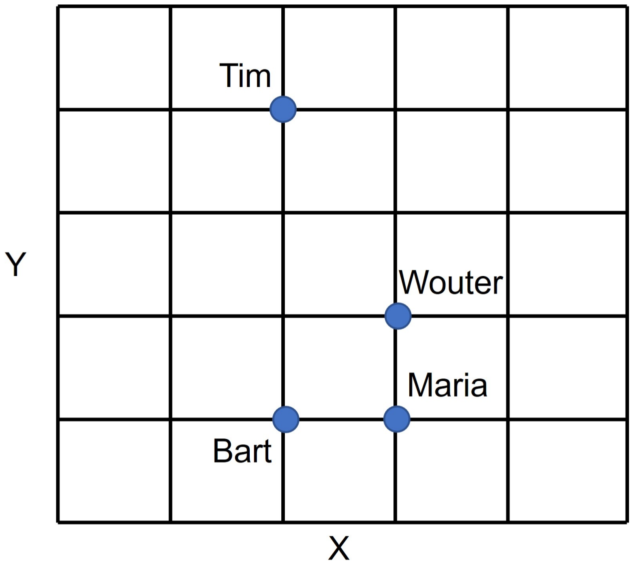


Figure 1.10: Local Outlier Factor (citatie naar PhD Eugen).

The K-distance is defined as the distance between observation and its K-th nearest neighbor. Hence, for $K = 2$, we have:

- K-distance Bart: 2
- K-distance Maria: 1
- K-distance: Wouter: 2
- K-distance Tim: 3

The K-neighbors of an observation A, $N_k(A))$, are the set of observations that lie in or on circle of radius K-distance measured using the Euclidean, Manhattan or other distance metric. Hence, we have:

- K-neighbors Bart: Maria and Wouter

- K-neighbors Maria: Bart, Wouter
- K-neighbors Wouter: Bart, Maria
- K-neighbors Tim: Bart, Wouter

The Reachability Distance (RD) is then defined as the maximum of the K-distance and the distance between two observations, or:

$$RD(x_i, x_j) = max(K - distance(x_i), distance(x_i, x_j)) \tag{1.9}$$

This gives:

- RD(Bart, Maria) = max(1, 1) = 1
- RD(Bart, Wouter) = max (2, 2) = 2
- RD(Bart, Tim) = max(3,3) = 3
- RD(Maria, Bart) = max(1,2) = 2
- RD(Maria, Wouter) = max(1,2) = 2
- RD(Maria, Tim) = max(4,3) = 4
- RD(Wouter, Bart) = max(2,2) = 2
- RD(Wouter, Maria) = max(1,1) = 1
- RD(Wouter, Tim) = max(3,3) = 3
- RD(Tim, Bart) = max(3,2) = 3
- RD(Tim, Maria) = max(4,1)= 4
- RD(Tim, Wouter) = max(3,2) = 3

The Local Reachability Density (LRD) is then defined as the inverse of average reachability distance of observation A from its neighbors or

$$LRD_k(A) = \frac{1}{\sum_{x_j \in N_k(A)} \frac{RD(A,X_j)}{|N_k(A)|}} \tag{1.10}$$

It specifies how far an observation is from the nearest cluster of observations. In other words, a lower value of $LRD_k(A)$ implies that the neighbors are far away from A and that there is less density around A. The calculations now become:

- LRD Bart 1/((1+2)/2) = 0,667
- LRD Maria: 1/((2+2)/2) = 0,50
- LRD Wouter: 1/((1+2)/2) = 0,667
- LRD Tim: 1/((3+3)/2) = 0,337

The Local Outlier Factor of observation A, $LOF(A)$, can now be calculated as:

$$LOF(A) = \frac{\sum_{x_j \in N_k(A)} LRD_k(X_j)}{|N_k(A)|} \times \frac{1}{LRD_k(A)} \tag{1.11}$$

In other words, $LOF(A)$ is the average LRD of all K neighbors of compared to the LRD of A. Hence, if $LOF(A)$ is close to 1, it indicates similar densities so A is not an anomaly or outlier. However, if $LOF(A)$ is bigger than 1, it implies that the LRD of A is less than the average LRD of the neighbors so A is likely to be an outlier. The idea is then sort all observations according to LOF and look at the biggest first for anamolous behaviour. In our example, the calculations become:

- LOF Bart: (0,5 + 0,667)/2 × 1/0,667 = 0,87
- LOF Maria: (0,667 + 0,667)/2 × 1/0,5 = 1,334
- LOF Wouter: (0,5 + 0,667)/2 × 1/0,667 = 0,87
- LOF Tim: (0,667 + 0,667)/2 × 1/0,3367 = 2

So in other words, Tim is the observation who is likely to be an outlier.

The key strength of LOF is that it is very successfully at detecting local outliers situated in dense or sparse data regions. It has no distributional assumptions and can work with varying densities. It also allows to rank observations in terms of their LOF score. A disadvantage is that a value for K must be chosen upfront which clearly impacts the number of distance calculations and hence computational complexity. It is also less powerful for high-dimensional data sets with many variables as distances become less meaningful then unless some unsupervised variable selection is done first.

1.5.7 One-Class SVMs

One class SVMs try to maximize the distance between a hyperplane and the origin [Schölkopf et al., 1999]. The idea is to separate the majority of the observations from the origin. The observations that lie on the other side of the hyperplane, closest to the origin, are then considered as anomalies. This is illustrated in Figure 1.11. One-class SVMs define a hyperplane as follows:

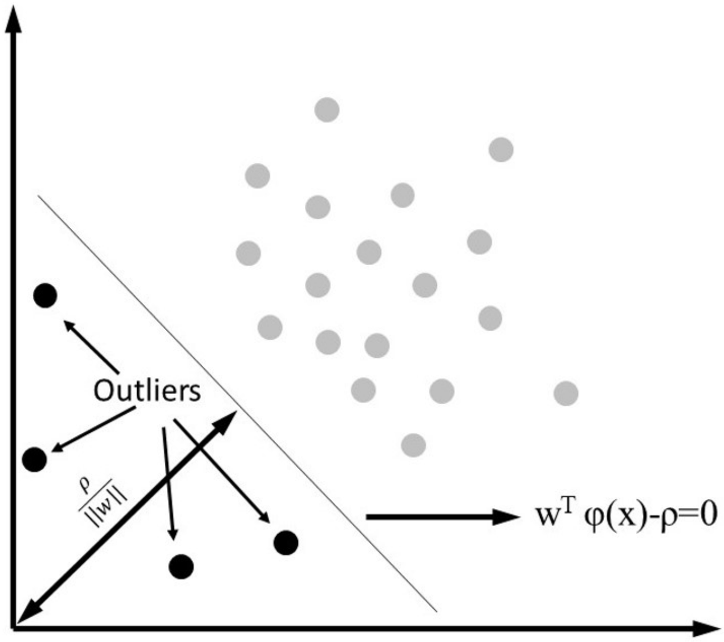


Figure 1.11: One-Class SVM.

$$w^T \phi(x) - \rho = 0 \tag{1.12}$$

Normal observations lie above the hyperplane and outliers below it, or in other words normal observations (outliers) will return a positive (negative) value for $f(x) = sign(w^T \phi(x) - \rho)$. Remember that $sign(x) = +1 if x > 0$ and -1 otherwise. One class SVMs then aim at solving the following optimization function:

$$\begin{aligned}&\text{Minimize}\tfrac{1}{2}\textstyle\sum_{i=1}^{N} w_i^2 - \rho + \tfrac{1}{\nu n}\textstyle\sum_{i=1}^{n} e_i\\&\text{subject to:}w^T\phi(x) \geq \rho - e_k, k = 1, \ldots, n\\&e_k \geq 0\end{aligned} \tag{1.13}$$

The error variables e_i are introduced to allow observations to lie on the side of the hyperplane closest to the origin. The parameter ν is a regularization term. Mathematically, it can be shown that the distance between the hyperplane and the origin equals $\frac{\rho}{||w||}$. This distance is then maximized by minimizing $\frac{1}{2}\sum_{i=1}^{N} w_i^2 - \rho$, which is the first part in the objective function. The second part of the objective function then accounts for errors, or thus outliers. The constraints force the majority of observations to lie above the hyperplane. The parameter ν ranges between 0 and 1, and sets an upper bound on the fraction of outliers. A lower (higher) value of the regularization parameter ν will increase (decrease) the weight assigned to errors and thus decrease (increase) the number of outliers. Given the importance of this parameter, one class SVMs are sometimes also referred to as ν-SVMs.

As with SVMs for supervised learning, the optimization problem can be solved by formulating its dual variant, which also here yields a quadratic programming (QP) problem, and applying the kernel trick. By again using Lagrangian optimization, the following decision function is obtained

$$f(x) = sign(w^T\phi(x) - \rho) = \text{sign}(\sum_{i=1}^{n} \alpha_i K(x, x_i) - \rho) \tag{1.14}$$

whereby α_i represent the Lagrange multipliers, and $K(x, x_i)$ the kernel function (for more details, see [Schölkopf et al., 1999]).

1.5.8 Other Methods

Some of the methods we discussed earlier can also be used for anomaly detection. One example is DBSCAN, discussed in Chapter **??**. As we explained, it makes a distinction between core, border and noise points with the latter obviously being the ones of interest in anomaly detection. Also deep learning methods such as auto-encoders and Generative Adversarial Networks (GANs) as discussed in Chapter XXX are interesting alternatives. When using autoencoders, observations that are hard to encode and hence come with high reconstruction errors are more likely to be anomalous. GANs have two competing neural network components: a generator and a discriminator. They are first trained on normal data, during which the generator learns to create synthetic data indistinguishable from the normal data, and the discriminator learns to differentiate between the two. When analyzing new data, if the discriminator identifies observations as significantly different from the normal data it has been trained on, these observations are likely to be considered outliers or anomalous.

1.5.9 Benchmarking anomaly detection techniques

In [Tiukhova et al., 2022] we compared iForest, DBSCAN and LOF techniques on 12 datasets with ground truth labels available (i.e., whether an observation was indeed anomalous or not) using five performance metrics: area under the precision recall curve (AUPR), F1 score, recall, precision and time (i.e., the sum of time to train and time to predict). Three main conclusions were drawn. First, the models considered disagree differently, i.e. their type I (observations incorrectly labeled as anomalies) and type II (observations incorrectly labelled as non-anomalies) errors are not similar. Second, considering the time, AUPRC and recall metrics, the iForest model is ranked the highest. Hence, the iForest model is the best in the cases when time performance is a key consideration as well as when the opportunity costs of not detecting an outlier are high. Third, the DBSCAN model obtains the highest ranking along the F1 score and precision dimensions. That allows us to conclude that if raising many false alarms is not an important concern, the DBSCAN model is the best to use.

1.6 CONCLUDING REMARKS

Pattern and anomaly discovery represent essential techniques in modern data analytics. Association and sequence rules help uncover meaningful relationships in transactional data, from market basket analysis to customer journey mapping, though careful attention must be paid to rule evaluation metrics and post-processing to identify truly valuable insights. The selection of appropriate minimum support and confidence thresholds remains critical but challenging, requiring iterative refinement and domain expertise.

Anomaly detection techniques have evolved from simple statistical approaches to sophisticated algorithms like isolation forests and one-class SVMs. Each method offers distinct advantages - LOF excels at detecting local outliers in varying density regions, while isolation forests provide computational efficiency and interpretable results. Recent research suggests that different techniques may be optimal depending on specific requirements: isolation forests when processing speed and recall are paramount, and DBSCAN when precision is the primary concern. As data volumes grow and patterns become more complex, combining multiple complementary approaches while considering their computational trade-offs will become increasingly important for robust anomaly detection.

Bibliography

R. Agrawal and S. Ramakrishnan. Fast algorithms for mining association rules. In *Proceedings of the 20th International Conference on Very Large Databases (VLDB)*, volume 1215, pages 487–499, 1994.

B. Baesens and A. De Caigny. *Customer Lifetime Value Modeling with Applications in Python and R*. Independently published, 2022.

B. Baesens, S. Viaene, and Vanthienen J. Post-processing of association rules. 2000a.

Bart Baesens, Stijn Viaene, and Jan Vanthienen. Post-processing of association rules. *Workshop on Post-Processing in Machine Learning and Data Mining: Interpretation, Visualization, Integration, and Related Topics in KDD*, 2000b.

Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422. IEEE, 2008.

Srikant Ramakrishnan and Rakesh Agrawal. Mining generalized association rules. In *Proceedings of the 21st International Conference on Very Large Databases (VLDB)*, pages 407–419. Morgan Kaufmann, 1995.

Bernhard Schölkopf, Robert Williamson, Alex Smola, John Shawe-Taylor, and John Platt. Support vector method for novelty detection. volume 12, pages 582–588, 01 1999.

Craig Silverstein, Sergey Brin, and Rajeev Motwani. Beyond market baskets: Generalizing association rules to dependence rules. *Data Mining and Knowledge Discovery*, 2:39–68, 1998. URL `https://api.semanticscholar.org/CorpusID:6532767`.

Ramakrishnan Srikant and Rakesh Agrawal. Mining quantitative association rules in large relational tables. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*. ACM, 1996.

E. Stripling, B. Baesens, B. Chizi, and S. vanden Broucke. Isolation-based conditional anomaly detection on mixed-attribute data to uncover workers' compensation fraud. *Decision Support Systems*, 111: 13–26, 2018. ISSN 0167-9236. doi: https://doi.org/10.1016/j.dss.2018.04.001. URL `https://www.sciencedirect.com/science/article/pii/S016792361830068X`.

Eugen Stripling. *Business-Oriented Data Analytics: Advances in Profit-Driven Model Building and Fraud Detection*. PhD thesis, Faculty of Economics and Business, KU Leuven, 2018.

Elena Tiukhova, Manon Reusens, Bart Baesens, and Monique Snoeck. Benchmarking conventional outlier detection methods. In Renata Guizzardi, Jolita Ralyté, and Xavier Franch, editors, *Research Challenges in Information Science*, pages 597–613, Cham, 2022. Springer International Publishing. ISBN 978-3-031-05760-1.